

Description

Method and arrangement for the computer-aided interchange of cryptographic keys between a first computer unit and a second computer unit

The invention relates to the computer-aided interchange of cryptographic keys between a first computer unit and a second computer unit.

Information technology systems are subject to various threats. Thus, by way of example, transmitted information can be tapped and modified by an unauthorized third party. A further threat during communication between two communication parties is that of a false identity of one communication party being feigned.

These and other threats are countered by various security mechanisms which are intended to protect the information technology system from the threats. One security mechanism used for safeguarding purposes is encryption of the transmitted data. To be able to encrypt the data in a communication link between two communication parties, steps which prepare the encryption first need to be taken before the actual data is transmitted. By way of example, the steps may involve the two communication parties agreeing to an encryption algorithm and, if appropriate, the common secret keys being declared.

The encryption security mechanism takes on particular significance in the case of mobile radio systems, since the transmitted data in these systems can be tapped by any third party without any particular additional effort.

This leads to the requirement for known security mechanisms to be selected and these security mechanisms

to be suitably combined, and also for communication protocols to be specified, such that they ensure the security of information technology systems.

- 5 Various asymmetric methods for the computer-aided interchange of cryptographic keys are known.

Asymmetric methods which are suitable for mobile radio systems are described in [1], [2], [3] and [4].

10

The method described in [1] relates expressly to local-area networks and makes relatively high demands in terms of computing power on a computer unit of a communication party during the key interchange.

15

Moreover, more transmission capacity is required in the method than in the method according to the invention, since the length of the messages is greater than in the case of the invention.

20

The method described in [2] has not implemented a few fundamental security aims. Explicit authentication of the network by the user is not achieved. Moreover, a key transmitted to the network by the user is not confirmed to the user by the network. There is also no assurance for the network that the key is fresh (up to date). A further disadvantage of this method is the restriction to the Rabin method in the implicit authentication of the key by the user. This restricts the method in terms of more flexible applicability. In

25

addition, no security mechanism which ensures the incontestability of transmitted data is provided. This is a considerable disadvantage, in particular also for the preparation of incontestable charge accounts for a mobile radio system. The restriction of the method to

30

the signature function used being the National Institute of Standards in Technology Signature Standard (NIST DSS) also restricts the method in its general applicability.

The method described in [3] has not implemented a fundamental security aim: explicit authentication of the user by the network is not achieved.

5 The method described in [4] is based on the assumption of the existence of common secret keys both between the user and the visited network and between the user and the home network before a protocol pass starts. This assumption is too restrictive for many instances of
10 use.

In addition, [5] discloses a method for secure data interchange between a multiplicity of subscribers involving a certification authority: The protocol used
15 for this method has a random number, an identity statement and also a public key and a session key. However, fundamental security aims are not implemented in this method.

20 In addition, [6] discloses a method for PC-PC communication involving a trust center.

[7] discloses a method in which a session key is produced using both a public key and a secret key and
25 also using a random number. This session key is combined with a public key.

In addition, [8] describes a method in which a user unit identifies itself to a network unit. An
30 authentication process then takes place between the user unit and the network unit using a hash function.

[9] discloses further secure communication protocols which nevertheless do not implement important
35 fundamental security aims.

[10] discloses the practice of forming a first value in a first computer unit from a first random number using

a generating element of a finite group, and transmitting it to a second computer unit. In the second computer unit, a session key is formed by hash value formation for the first value, which is 5 exponentiated using a secret network key. The session key is likewise formed in the first computer unit, but there by hash value formation for a public network key which is exponentiated using the first random number. In addition, a hash value for the session key is formed 10 there and the hash value is digitally signed. The resultant signature term is transmitted to the second computer unit and is verified there.

15 The method described in [11] achieves the important security aims, but with a relatively high level of input in terms of computing power and transmission capacity.

20 Asymmetric methods are essentially based on two complexity theory problems, the problem of efficiently factorizing composed numbers and the discrete logarithm problem (DLP). The DLP is that, although exponentiation operations can be carried out efficiently in suitable computing structures, no efficient algorithms are known 25 for the reversal of this operation, logarithmation.

By way of example, the finite groups referred to above are to be understood as being such computing structures. These groups are, for example, the 30 multiplicative group of a finite body (e.g. multiplication modulo p, where p is a large prime number), or else so-called "elliptical curves". Elliptical curves are primarily of interest because they permit much shorter security parameters for the 35 same level of security. This relates to the length of the public keys, to the length of the certificates, to the length of the messages to be interchanged during session key declaration and to the length of digital

signatures, which are each described below. The reason for this is that the logarithmation methods known for elliptical curves are much less efficient than those for finite bodies.

5

In this context, a large prime number means that the size of the prime number needs to be selected such that logarithmation is so complex that it cannot be performed in a reasonable time. In this context, reasonable means a period of time corresponding to the security policy over a number of years to decades, and longer.

15 In this context, a hash function is to be understood as being a function in the case of which it is not possible to calculate a matching input value for a given function value. In addition, an input character sequence of arbitrary length is allocated an output character sequence of fixed length. Furthermore, 20 additional properties may be demanded for the hash function. One such additional property is freedom from conflict, i.e. it must not be possible to find two different input character sequences which produce the same output character sequence.

25 ~~aa~~

The invention is based on the problem of specifying a simplified method for the computer-aided interchange of cryptographic keys which does not presuppose the existence of common secret keys.

30

This problem is solved by the method according to patent claim 1 and by the arrangement according to patent claim 29.

35 In the method, a first value is formed from a first random number using a generating element of a finite group in the first computer unit. A first message is transmitted from the first computer unit to the second

computer unit, the first message containing at least the first value. A session key is formed in the second computer unit using a first hash function, a first input variable for the first hash function containing
5 at least one first term which is formed by exponentiation of the first value using a secret network key. The session key is formed in the first computer unit using the first hash function, a second input variable for the first hash function containing
10 at least one second term which is formed by exponentiation of a public network key using the first random number. A fourth input variable is formed in the first computer unit using a second hash function or the first hash function, a third input variable for the
15 first hash function or for the second hash function containing, for the purpose of forming the fourth input variable, one or more variables which can be used to infer the session key unambiguously. A signature term is formed in the first computer unit from at least the
20 fourth input variable using a first signature function. A third message is transmitted from the first computer unit to the second computer unit, the third message containing at least the signature term from the first computer unit. The signature term is verified in the
25 second computer unit.

In the case of the arrangement, the first computer unit and the second computer unit are set up such that the following method steps can be carried out:

- 30 - a first value is formed from a first random number using a generating element of a finite group in the first computer unit,
- a first message is transmitted from the first computer unit to the second computer unit, the first message containing at least the first value,
35 - a session key is formed in the second computer unit using a first hash function, a first input variable for the first hash function containing at least one

- first term which is formed by exponentiation of the first value using a secret network key,
- the session key is formed in the first computer unit using the first hash function, a second input variable for the first hash function containing at least one second term which is formed by exponentiation of a public network key using the first random number,
 - a fourth input variable is formed in the first computer unit using a second hash function or the first hash function, a third input variable for the first hash function or for the second hash function containing, for the purpose of forming the fourth input variable, one or more variables which can be used to infer the session key unambiguously,
 - a signature term is formed in the first computer unit from at least the fourth input variable using a first signature function,
 - a third message is transmitted from the first computer unit to the second computer unit, the third message containing at least the signature term from the first computer unit, and
 - the signature term is verified in the second computer unit.

The advantages which are achieved by the invention are primarily a considerable reduction in the length of the transmitted messages and the implementation of further security aims.

In addition, the invention can be adapted very easily to different requirements, since there is no restriction to particular algorithms for signature formation and encryption.

Advantageous developments of the invention can be found in the dependent claims.

In one development, provision is made for a long-service secret network key and a long-service public network key to be used.

- 5 A long-service key is to be understood below as being a key which is used for a plurality of protocol passes.

The invention and its developments implement the following security aims:

- 10 - mutual explicit authentication by the user and the network, i.e. mutual verification of the claimed identity,
- key declaration between the user and the network with mutual implicit authentication, i.e. the method
- 15 achieves the effect that, after completion of the procedure, a common secret session key is available, of which each party knows that only the authentic counterpart can likewise be in possession of the secret session key,
- 20 - assurance for the user that the session key is fresh (up to date),
- mutual confirmation of the session key by the user and the network, i.e. confirmation that the counterpart is actually in possession of the declared
- 25 secret session key.

The following advantageous developments of the method also relate to these security aims.

- 30 In one development, a dependable public user key for the first computer unit, e.g. in the form of a user certificate, is additionally made available in the first computer unit and a dependable public network key for the second computer unit, e.g. in the form of a
- 35 network certificate, is made available in the second computer unit. The public network key need not be available in the first computer unit in this development.

In a further refinement, it is not necessary for the public user key to be available in the second computer unit.

5 In accordance with a further refinement, no dependable public network key for the second computer unit is necessary in the first computer unit. A dependable public certification key for the certification computer unit is available in the first computer unit. This
10 means that the first computer unit needs to "acquire" the dependable public network key in the form of a network certificate from a certification computer unit. The second computer unit likewise needs the dependable public user key in the form of a user certificate from
15 the certification computer unit.

The developments of the invention in accordance with patent claims 13, 15 and 20 implement the security aim of user anonymity, i.e. confidentiality of the identity
20 of the user with regard to third parties.

The development of the method according to the invention in accordance with patent claim 15 permits the use of temporary user identities.

25 The development of the method in accordance with patent claim 16 primarily ensures additional authentication of the second computer unit with regard to the first computer unit.

30 The development in accordance with patent claim 18 implements the security aim of assurance for the network that the session key is fresh (up to date).

35 The development in accordance with patent claim 21 additionally implements the security aim of incontestability of data which has been sent from the user to the network.

The drawings show preferred exemplary embodiments of the invention which are described in more detail below.

In the drawings.

- 5 Figure 1 shows a flowchart illustrating a first exemplary embodiment of the method with a few developments;
- 10 Figure 2 shows a flowchart illustrating a second exemplary embodiment of the method with a few developments;
- 15 Figure 3 shows a flowchart illustrating a third exemplary embodiment of the method with a few developments.

15 First exemplary embodiment

Figure 1 shows a sketch of the execution of the method. The method relates to the interchange of cryptographic keys between a first computer unit U and a second computer unit N, where the first computer unit U is to be understood as being a computer unit of a user of a mobile radio network and a second computer unit N is to be understood as being a computer unit of the network operator of a mobile radio system.

It is a prerequisite for the method that a dependable public network key g^s for the second computer unit N is available in the first computer unit U and that a dependable public user key KU for the first computer unit U is available in the second computer unit N, where g is a generating element of a finite group.

In the first computer unit U, a first random number t is generated (step 101). The generating element g of a finite group is used to form a first value g^t from the first random number t in the first computer unit U (step 102).

Once the first value g^t has been calculated, a first message M₁, containing at least the first value g^t , is coded. The first message M₁ is transmitted from the first computer unit U to the second computer unit N
5 (step 103).

In the second computer unit N, the first message M₁ is decoded. The first message M₁ may also be transmitted over an insecure channel, that is to say also via an
10 air interface, in unencrypted form, since the logarithmation of the first value g^t cannot be performed in a reasonable time.

15 In the second computer unit N, a second random number r is generated (step 104). This additional method step implements an additional security aim: the assurance for the second computer unit N that a session key K described below is fresh (up to date).

20 In the second computer unit N, a first hash function h₁ is used to form a session key K (step 105). At least one first term is used as a first input variable for the first hash function h₁. The first term is formed by raising the first value g^t to a higher power using a
25 secret network key s.

If the second random number r is used, the first input variable for the first hash function h₁ additionally contains at least the second random number r.

30 A response A is now formed in the second computer unit N (step 106). Various variants are provided for forming the response A. Thus, for example, it is possible for an encryption function Enc to be used to encrypt a
35 constant const, and possibly further variables, with the session key K. The constant const is known both to the first computer unit U and to the second computer unit N. The encryption function Enc is also known both

to the second computer unit N and to the first computer unit U as the encryption function which is to be used in the method.

5 A further option for forming the response A (step 106) is for the session key K, and possibly prescribable further variables, e.g. an identity statement idN for the second computer unit N and/or the second random number, to be used as input variable for a second hash
10 function h2, and for the "hashed" value for the session key K, and possibly for the further variables, to be used as response A.

Stringing together the second random number r, the
15 response A and also an optional first data field dat1 forms a second message M2. The optional first data field dat1 is only contained in the second message M2 if this is provided in the method.

20 The second message M2 is coded in the second computer unit N and is transmitted to the first computer unit U (step 107).

In the first computer unit U, the second message M2 is
25 decoded, which means that the first computer unit U has the second random number r, the response A and possibly the optional first data field dat1 available. The length of the optional first data field dat1 may be of any desired size, i.e. it is also possible for the
30 optional first data field dat1 not to be present.

In the first computer unit U, the session key K is now likewise formed (step 108), using the first hash function h1, which is known both to the second computer unit N and to the first computer unit U. A second input variable for the first hash function h1 for forming the session key K in the first computer unit U contains at least one second term. The second term is formed from

exponentiation of a public network key g^s using the first random number t . If the use of the second random number r is provided in the method for calculating the session key K , the second input variable for the first 5 hash function h_1 for forming the session key K in the first computer unit U additionally contains the second random number r .

The use of the first random number t and of the second 10 random number r for generating the session key K ensures that the session key K is up to date, since the first random number t and the second random number r are respectively used only for one session key K in each case. This prevents reinjection of an older key as 15 the session key K .

Once the session key K has been formed in the first computer unit U , the received response A is used to check whether the session key K formed in the first 20 computer unit U matches the session key K which was formed in the second computer unit N (step 109). Subject to the variants described above for forming the response A , various options are provided for checking the session key K using the response A .

25 One option is that, if the response A has been formed in the second computer unit N by encrypting the constant $const$, and possibly further variables, with the session key K using the encryption function Enc , the response A is decrypted, and hence the first 30 computer unit U receives a decrypted constant $const'$, and possibly prescribable further variables, which is/are compared with the known constant $const$, and possibly the further variables.

35 The session key K may also be checked, using the response A , by encrypting the constant $const$, known to the first computer unit U , and possibly prescribable

further variables, with the session key K, formed in the first computer unit U, using the encryption function Enc and checking the result with the response A for a match. This procedure is also used when the 5 response A is formed in the second computer unit N, by applying the second hash function h2 to the session key K, and possibly to the further variables. In this case, the session key K formed in the first computer unit U, and possibly prescribable further variables, is/are 10 used as input variable for the second hash function h2 in the first computer unit U. The "hashed" value for the session key K formed in the first computer unit U, and possibly for further variables, is then checked with the response A for a match. This achieves the aim 15 of key confirmation for the session key K.

As a result of the secret network key s being used for calculating the session key K in the second computer unit N, and the public network key g^s being used for calculating the session key K in the first computer unit U, the second computer unit N is authenticated by the first computer unit U. This is achieved provided that it is known for the first computer unit U that the 20 public network key g^s actually belongs to the second computer unit N. 25

Subsequent to confirmation of the session key K by means of a check on the response A, a signature term is calculated (step 110). To this end, a third hash 30 function h3 is used to form a fourth input variable. The third hash function h3 can, but need not, be the same hash function as the first hash function h1 and/or the second hash function h2. As a third input variable for the third hash function h3, a term is used which 35 contains one or more variables from which it is possible to infer the session key unambiguously. In addition, the third input variable may contain the optional first data field dat1 or else an optional

second data field dat2, if the use thereof is provided in the method.

Such variables are the first value g^t , the public network key g^s and the second random number r.
5

It is subsequently not possible to contest the fact that the data contained in the first optional data field dat1 and in the second optional data field dat2
10 has been sent from the first computer unit U.

The data contained in the first optional data field dat1 and in the second optional data field dat2 may be telephone numbers, the current time or similar
15 parameters suitable for this purpose. This information may be used as a tool for incontestable charge accounting.

A first signature function Sig_u is used to form the
20 signature term from at least the fourth input variable. To achieve a higher degree of security, the signature term may be encrypted. In this case, the signature term is encrypted with the session key K using the encryption function Enc and forms the first encrypted
25 term VT1.

In addition, if the security aim of "anonymity of the user" is to be implemented, a second encrypted term VT2
30 is calculated by encrypting an identity variable IMUI for the first computer unit U with the session key K using the encryption function Enc. When an optional second data field dat2 is used, a third encrypted term VT3 is calculated in the first computer unit U by encrypting the optional second data field dat2 with the
35 session key K using the encryption function Enc; the optional second data field dat2 may also be transmitted in unencrypted form.

The three encrypted terms may also be combined to form a fourth encrypted term VT4, in which the interlinkage of signature term, identity variable IMUI and optional second data field dat2 is encrypted with the session key K (step 111).

In the first computer unit U, a third message M3, containing at least the signature term and the identity variable IMUI for the first computer unit U, is formed and coded.

If anonymity of the first computer unit U is to be ensured, the third message M3 contains, instead of the identity variable IMUI for the first computer unit U, at least either the second encrypted term VT2 or the fourth encrypted term VT4, which contains the information about the identity of the first computer unit U in encrypted form, which can be decrypted only by the second computer unit N.

If the use of the optional second data field dat2 is provided, the third message M3 additionally contains at least the third encrypted term VT3 or the fourth encrypted term VT4 or the optional second data field dat2 in plain text.

If the third message M3 contains the first encrypted term VT1, the second encrypted term VT2 or the third encrypted term VT3 or the fourth encrypted term VT4, these are decrypted in the second computer unit N. This is done for the first encrypted term VT1, which may be present, before verification of the signature term.

The third message M3 is transmitted from the first computer unit U to the second computer unit N (step 112).

In addition, authentication of the first computer unit U for the second computer unit N is ensured by the signature term, which contains the random number r, the use of which guarantees that the third message M3 has 5 actually been sent from the first computer unit U at the present time.

In the second computer unit N, the third message M3 is decoded, decrypted, and a user certificate CertU 10 available to the second computer unit N is then used to verify the signature term (step 113).

If temporary user identities are provided for the method, then the method described above is extended by 15 a few method steps.

The second computer unit N must first be informed of which first computer unit U is to be allocated a new temporary identity variable TMUIN by the second 20 computer unit N.

To this end, an old temporary identity variable TMUIO is transmitted from the first computer unit U to the second computer unit N as an additional component of 25 the first message M1.

Once the first message M1 has been received, the second computer unit N thus knows for which first computer unit U the new temporary identity variable TMUIN is 30 intended.

The new temporary identity variable TMUIN for the first computer unit U is then formed in the second computer unit N. This may be performed, for example, by 35 generating a random number or by means of tables in which potential identity variables are stored. The new temporary identity variable TMUIN for the first computer unit U is used to form a fifth encrypted term

VT5 in the second computer unit N by encrypting the new temporary identity variable TMUIN for the first computer unit U with the session key K using the encryption function Enc.

5

In this case, the second message M2 additionally contains at least the fifth encrypted term VT5. The fifth encrypted term VT5 is then decrypted in the first computer unit U. The new temporary identity variable TMUIN for the first computer unit U is now available in the first computer unit U.

So that the second computer unit N is also assured of the fact that the first computer unit U has received the new temporary identity variable TMUIN correctly, the third input variable for the first hash function h1 or for the third hash function h3 additionally contains at least the new temporary identity variable TMUIN for the first computer unit U.

20

Since the information for the new temporary identity variable TMUIN is contained in the signature term in this case, the third message M3 no longer contains the identity variable IMUI for the first computer unit U.

25

It is also possible for the new temporary identity variable TMUIN not to be integrated into the signature term, but rather for the second encrypted term VT2 to be formed by encrypting, instead of the identity variable IMUI for the first computer unit U, the new temporary identity variable TMUIN with the session key K using the encryption function Enc. In this case, the third message M3 additionally contains the second encrypted term VT2.

35

The hash functions used in the method, the first hash function h1, the second hash function h2 and the third

hash function h_3 can be produced by the same hash functions, or else by different hash functions.

Second exemplary embodiment

5 *Sur*
① Figure 2 shows a sketch of the execution of a second exemplary embodiment of the method.

10 A prerequisite for this exemplary embodiment of the method is that a dependable public user key K_U for the first computer unit U in the form of a user certificate $Cert_U$ is made available in the first computer unit U , and that a dependable public network key g^s for the second computer unit N in the form of a network certificate $Cert_N$ is made available in the second computer unit N . The public network key g^s need not be available in the first computer unit U . Likewise, it is not necessary for the public user key K_U to be available in the second computer unit N .

20 In the first computer unit U , the first random number t is generated (step 201). The generating element g of a finite group in the first computer unit U is used to form the first value g^t from the first random number t (step 202).

Once the first value g^t has been calculated, a first message M_1 is coded, said first message containing at least the first value g^t and an identity statement id_{CA} 30 for a certification computer unit CA which delivers the network certificate $Cert_N$ which can be verified by the first computer unit U . The first message M_1 is transmitted from the first computer unit U to the second computer unit N (step 203).

35 In the second computer unit N , the first message M_1 is decoded.

As described in figure 2, a second random number r is generated in the second computer unit N (step 204). This additional method step implements an additional security aim: the assurance for the second computer unit N that a session key K described below is fresh (up to date).

In the second computer unit N, the first hash function h_1 is used to form the session key K (step 205). The first input variable used for the first hash function h_1 is a first term. The first term is formed by raising the first value g^t to a higher power using the secret network key s.

When the second random number r is used, the first input variable for the first hash function h_1 additionally contains at least the second random number r .

A response A is now formed in the second computer unit N (step 206). To form the response A, the variants described within the context of the first exemplary embodiment are provided.

Stringing together the second random number r , the network certificate CertN, the response A and an optional first data field dat1 forms the second message M2. The optional first data field dat1 is only contained in the second message M2 if this is provided in the method.

The second message M2 is coded in the second computer unit N and is transmitted to the first computer unit U (step 207).

In the first computer unit U, the second message M2 is decoded, which means that the first computer unit U has the second random number r , the response A and possibly

the optional first data field dat1 available. The length of the optional first data field dat1 can be of any desired size, i.e. it is also possible for the optional first data field dat1 not to be present.

5

Next, the network certificate CertN contained in the second message M2 is verified in the first computer unit. Hence, the public network key g^s is available in the first computer unit U.

10

In the first computer unit U, the session key K is now likewise formed (step 208), using the first hash function h1, which is known both in the second computer unit N and in the first computer unit U. A second input variable for the first hash function h1 for forming the session key K in the first computer unit U contains at least one second term. The second term is formed from exponentiation of the public network key g^s using the first random number t. If the use of the second random number r is provided in the method for calculating the session key K, the second input variable for the first hash function h1 for forming the session key K in the first computer unit U additionally contains the second random number r.

25

The use of the first random number t and of the second random number r for generating the session key K ensures that the session key K is up to date, since the first random number t and the second random number r are respectively used only for one session key K in each case. This prevents reinjection of an older key as the session key K.

Once the session key K has been formed in the first computer unit U, the received response A is used to check whether the session key K formed in the first computer unit U matches the session key K which was formed in the second computer unit N (step 209).

Subject to the variants described above for forming the response A, various options are provided for checking the session key K using the response A.

- 5 To check the response A, the variants described within the context of the first exemplary embodiment are provided. This achieves the aim of key confirmation for the session key K.
- 10 As a result of the secret network key s being used for calculating the session key K in the second computer unit N, and the public network key g^s being used for calculating the session key K in the first computer unit U, the second computer unit N is authenticated by
- 15 the first computer unit U. This is achieved provided that it is known for the first computer unit U that the public network key g^s actually belongs to the second computer unit N.
- 20 Subsequent to confirmation of the session key K by means of a check on the response A, the signature term is calculated (step 210). To this end, the third hash function h3 is used to form a fourth input variable. The third hash function h3 can, but need not, be the
- 25 same hash function as the first hash function h1 and/or the second hash function h2. As a third input variable for the third hash function h3, a term is used which contains one or more variables from which it is possible to infer the session key unambiguously. In
- 30 addition, the third input variable may contain the optional first data field dat1 or else an optional second data field dat2, if the use thereof is provided in the method.
- 35 Such variables are the first value g^t, the public network key g^s and the second random number r.

It is subsequently not possible to contest the fact that the data contained in the first optional data field dat1 and in the second optional data field dat2 has been sent from the first computer unit U.

5

The data contained in the first optional data field dat1 and in the second optional data field dat2 may be telephone numbers, the current time or similar parameters suitable for this purpose. This information 10 may be used as a tool for incontestable charge accounting.

A first signature function Sig_u is used to form the signature term from at least the fourth input variable. 15 To achieve a higher degree of security, the signature term may be encrypted. In this case, the signature term is encrypted with the session key K using the encryption function Enc and forms the first encrypted term VT1.

20

In addition, if the security aim of "anonymity of the user" is to be implemented, a second encrypted term VT2 is calculated by encrypting a user certificate CertU for the first computer unit U with the session key K 25 using the encryption function Enc. When an optional second data field dat2 is used, a third encrypted term VT3 can be calculated in the first computer unit U by encrypting the optional second data field dat2 with the session key K using the encryption function Enc. The 30 optional second data field dat2 may likewise be transmitted in unencrypted form.

The three encrypted terms may also be combined to form a fourth encrypted term VT4, in which the chain comprising signature term, identity variable IMUI and optional second data field dat2 is encrypted with K (step 211).

In the first computer unit U, a third message M3, containing at least the signature term and the user certificate CertU for the first computer unit U, is formed and coded. If user anonymity of the first
5 computer unit U is to be ensured, the third message M3 contains, instead of the user certificate CertU for the first computer unit U, at least either the second encrypted term VT2 or the fourth encrypted term VT4, which contains the user certificate CertU for the first
10 computer unit U in encrypted form, which can be decrypted only by the second computer unit N.

If the use of the optional second data field dat2 is provided, the third message M3 additionally contains at
15 least the third encrypted term VT3 or the fourth encrypted term VT4. If the third message M3 contains the first encrypted term VT1, the second encrypted term VT2 or the third encrypted term VT3 or the fourth encrypted term VT4, these are decrypted in the second computer unit N. This is done for the first encrypted term VT1, which may be present, before verification of
20 the signature term.

The third message M3 is transmitted from the first
25 computer unit U to the second computer unit N (step 212).

In the second computer unit N, the third message M3 is decoded, decrypted, and a user certificate CertU available to the second computer unit N is then used to verify the signature term (step 213).

In addition, authentication of the first computer unit U for the second computer unit N is ensured by the
35 signature term, which contains the random number r, the use of which guarantees that the third message M3 has actually been sent from the first computer unit U at the present time.

If temporary user identities are provided for the method, then the method described above is extended by a few method steps.

- 5 In the second computer unit N, a new temporary identity variable TMUIN is formed for the first computer unit U and is subsequently allocated to the first computer unit U. This may be performed by generating a random number or by means of tables in which potential
10 identity variables are stored. The new temporary identity variable TMUIN for the first computer unit U is used to form a fifth encrypted term VT5 in the second computer unit N by encrypting the new temporary identity variable TMUIN for the first computer unit U
15 with the session key K using the encryption function Enc.

In this case, the second message M2 additionally contains at least the fifth encrypted term VT5. The
20 fifth encrypted term VT5 is then decrypted in the first computer unit U. The new temporary identity variable TMUIN for the first computer unit U is now available in the first computer unit U.

- 25 So that the second computer unit N is also assured of the fact that the first computer unit U has received the new temporary identity variable TMUIN correctly, the third input variable for the first hash function h1 or for the third hash function h3 additionally contains
30 at least the new temporary identity variable TMUIN for the first computer unit U.

It is also possible for the new temporary identity variable TMUIN not to be integrated into the signature term, but rather for the second encrypted term VT2 to be formed by encrypting the new temporary identity variable TMUIN for the first computer unit U with the session key K using the encryption function Enc. In

this case, the third message M3 additionally contains the second encrypted term VT2.

Third exemplary embodiment

5

Syst as Figure 3 shows a sketch of the execution of a third exemplary embodiment.

A prerequisite for this exemplary embodiment of the method is that no dependable public network key g^s for the second computer unit N is available in the first computer unit U. A dependable public certification key cs for a certification computer unit CA is available in the first computer unit U. This means that the first computer unit U needs to "acquire" the dependable public network key g^s in the form of a network certificate CertN from the certification computer unit CA. Likewise, the second computer unit N needs the dependable public user key KU in the form of a user certificate CertU from the certification computer unit CA.

In the first computer unit U, the first random number t is generated (step 301). The generating element g of a finite group in the first computer unit U is used to form the first value g^t from the first random number t (step 302).

Once the first value g^t has been calculated, a first message M1 is coded, said first message containing at least the first value g^t , an identity variable IMUI for the first computer unit U and an identity statement id_{CA} for a certification computer unit CA which delivers a network certificate CertN which can be verified by the first computer unit U. This is necessary when a plurality of certification authorities with different secret certification keys are provided. If the security aim of user anonymity is to be implemented, an

intermediate key L is formed in the first computer unit U before formation of the first message M1. This is done by raising the public key declaration key g^u for the certification computer unit CA, which key is available in the first computer unit U, to a higher power using the first random number t. Subsequently, the identity variable IMUI for the first computer unit U is in this case encrypted with the intermediate key L using an encryption function Enc, and the result represents a fifth encrypted term VT5. The fifth encrypted term VT5 is integrated into the first message M1 instead of the identity variable IMUI for the first computer unit U. The first message M1 is transmitted from the first computer unit U to the second computer unit N (step 303).

In the second computer unit N, the first message M1 is decoded and a fourth message M4 is formed (step 304), said fourth message containing a chain comprising the certificate CertN, known to the second computer unit N, for the public network key g^s , the first value g^t and the identity variable IMUI for the first computer unit U. If the security aim of user anonymity is to be implemented, the fifth encrypted term VT5 is coded in the fourth message M4 instead of the identity variable IMUI for the first computer unit U.

The fourth message M4 is coded in the second computer unit N and is then transmitted to the certification computer unit CA (step 304).

The fourth message M4 is decoded in the certification computer unit CA.

Next, if user anonymity is ensured, that is to say the fifth encrypted term VT5 has also been sent in the fourth message M4, the intermediate key L is calculated in the certification computer unit CA by raising the

first value g^t to a higher power using a secret key declaration key u for the certification computer unit CA.

5 The fifth encrypted term VT5 is decrypted with the intermediate key L using the encryption function Enc, as a result of which the identity variable IMUI for the first computer unit U is known in the certification computer unit CA.

10

In the certification computer unit CA, the user certificate CertU is then ascertained. The user certificate CertU is ascertained from a dedicated database for the certification computer unit CA, said 15 database containing all the certificates for the computer units for which the certification computer unit CA produces certificates.

20 To check the validity of the network certificate CertN and of the user certificate CertU, an identity statement id_N for the network computer unit N and the public network key g^s also sent in the fourth message, the identity variable IMUI for the first computer unit U and also the ascertained user certificate CertU are 25 compared with a revocation list containing invalid certificates, keys or identity variables.

The certification computer unit CA then forms three chains of certificates, a first certificate chain 30 CertChain (U, N), a second certificate chain CertChain (N, U) and a third certificate chain CertChain (N, CA).

35 The first certificate chain CertChain (U, N) can be verified by the first computer unit U using the public certification key for the certification computer unit CA, which is known to the first computer unit U, and contains as last element a certificate CertN for the public key g^s from the second computer unit N.

The second certificate chain CertChain (N, U) can be verified by the second computer unit N and contains as last element a certificate CertU for the public key KU from the first computer unit U.

5 The third certificate chain CertChain (N, CA) can be verified by the second computer unit N and contains as last element a certificate for the public verification 10 key from the certification computer unit CA.

The first certificate chain CertChain (U, N) and the second certificate chain CertChain (N, U) can be uniquely identified by the identifiers cidU and cidN.

15 Next, a third term is formed from at least one chain comprising the first value g^t and the identifiers cidU and cidN.

20 The third term is "hashed" using a fourth hash function h4, and the result of the hash function h4 is signed using a third signature function Sig_{CA} .

25 In addition, a time stamp TS is created in the certification computer unit CA. This time stamp is optionally included in the third term.

A fifth message M5 formed in the certification computer 30 unit CA contains at least one chain comprising the signed third term and the certificate chains CertChain (U, N) and CertChain (N, U), and also optionally the time stamp TS and the certificate chain CertChain (N, CA). The signed hash value for the third term and also the certificate chain CertChain (N, U) are optionally 35 encrypted using the intermediate key L.

The fifth message M5 is coded in the certification computer unit CA and is transmitted to the second

computer unit N (step 305). Once the fifth message M5 is decoded in the second computer unit N, the signed hash value for the third term is verified, provided that it is not encrypted with L.

5

In the second computer unit N, a fourth term is now formed, said fourth term containing at least one chain comprising the certificate chain CertChain (U, N) and the signed hash value (optionally encrypted with the intermediate key L) for the third term.

In the second computer unit N, the first hash function h1 is used to form a session key K. A first input variable used for the first hash function h1 is a concatenation of a first term with the second random number r. The first term is formed by raising the first value g^t to a higher power using a secret network key s. The second random number r is used when the intention is to implement the additional security aim of assurance for the second computer unit N that the session key K is fresh (up to date). If this security aim is not required, the second random number r is not used in the method for calculating the session key K.

25 In the second computer unit N, a response A is formed. For forming the response A, the variants described in the first exemplary embodiment are provided.

Stringing together the second random number r, the fourth term, the response A and also an optional first data field dat1 and the optional time stamp forms a second message M2. The optional first data field dat1 is only contained in the second message M2 if this is provided in the method.

35

The second message M2 is coded in the second computer unit N and is transmitted to the first computer unit U (step 306).

- In the first computer unit U, the second message M2 is decoded, which means that the first computer unit U has the second random number r, the response A and also possibly the optional first data field dat1 and
- 5 possibly the time stamp TS available. The length of the optional first data field dat1 can be of any desired size, i.e. it is also possible for the optional first data field dat1 not to be present.
- 10 In the first computer unit U, the session key K is now likewise formed (step 307), using the first hash function h1, which is known both to the second computer unit N and to the first computer unit U. A second input variable for the first hash function h1 for forming the
- 15 session key K in the first computer unit U contains at least one second term. The second term is formed from exponentiation of a public network key g^s using the first random number t. If the second random number r is provided in the method for calculating the session key
- 20 K, the second input variable for the first hash function h1 for forming the session key K in the first computer unit U additionally contains the second random number r.
- 25 Once the session key K has been formed in the first computer unit U, the received response A is used to check whether the session key K formed in the first computer unit U matches the session key K which was formed in the second computer unit N (step 308).
- 30 Subject to the variants described above for forming the response A, the options described above are provided for checking the session key K using the response A.
- 35 As a result of the secret network key s being used for calculating the session key K in the second computer unit N, and the public network key g^s being used for calculating the session key K in the first computer

unit U, the second computer unit N is authenticated by the first computer unit U. This is achieved provided that it is known for the first computer unit U that the public network key g^s actually belongs to the second computer unit N. That is achieved by U as a result of verification of the certificate chain CertChain (U, N) and also of the signed hash value for the third term. If the latter is encrypted with the intermediate key L, it needs to be decrypted using the intermediate key L before verification.

Subsequent to confirmation of the session key K by means of a check on the response A, a signature term is calculated (step 309). To this end, a third hash function h_3 is used to form a fourth input variable. The third hash function h_3 can, but need not, be the same hash function as the first hash function h_1 and/or the second hash function h_2 . As a third input variable for the third hash function h_3 , a term is used which contains one or more variables from which it is possible to infer the session key unambiguously. In addition, the third input variable may contain the optional first data field dat1 or else an optional second data field dat2, if the use thereof is provided in the method.

Such variables are the first value g^t , the public network key g^s and the second random number r.

- 30 It is subsequently not possible to contest the fact that the data contained in the first optional data field dat1 and in the second optional data field dat2 is sent from the first computer unit U.
- 35 The data contained in the first optional data field dat1 and in the second optional data field dat2 may be telephone numbers, the current time or similar parameters suitable for this purpose. This information

may be used as a tool for incontestable charge accounting.

A first signature function Sig_v is used to form the
5 signature term from at least the fourth input variable.
To achieve a higher degree of security, the signature
term may be encrypted. In this case, the signature term
is encrypted with the session key K using the
10 encryption function Enc and forms the first encrypted
term VT_1 .

When an optional second data field dat_2 is used, a
third encrypted term VT_3 is calculated in the first
15 computer unit U by encrypting the optional second data
field dat_2 with the session key K using the encryption
function Enc . The optional second data field dat_2 may
also be transmitted in unencrypted form, that is to say
in plain text.

20 As an alternative for forming the first and the third
encrypted term VT_1 and VT_3 , it is also possible for a
fourth encrypted term VT_4 to be formed by encrypting at
least the chain comprising the signature term and
optionally the data field dat_2 and the intermediate key
25 L using the session key K (step 310).

In the first computer unit U , a third message M_3 is
formed and coded, said third message comprising at
least the first encrypted term VT_1 and, if the optional
second data field dat_2 is used, the third encrypted
30 term VT_3 or the optional second data field dat_2 in
plain text, or else comprising the fourth encrypted
term VT_4 .

The third message M_3 is transmitted from the first
35 computer unit U to the second computer unit N (step
311).

In the second computer unit N, the third message M3 is decoded and then the first encrypted term VT1 and also possibly the third encrypted term VT3, or else the fourth encrypted term VT4, is decrypted. If parts of 5 the message M5 have been encrypted with L, then the second computer unit N can now use the intermediate key L received in message M3 to decrypt the encrypted parts of the message M5. The second computer unit N can then verify the second certificate chain Cert (N, U) and 10 also the signed hash value for the third term using the public verification key of CA. The user certificate CertU, which is now available to the second computer unit N, is used to verify the signature term.

15 In addition, authentication of the first computer unit U for the second computer unit N is ensured by the signature term in the third message M3, which contains the random number r, the use of which also guarantees that the third message M3 has actually been sent from 20 the first computer unit U at the present time.

If temporary user identities are provided for the method, then the method described above is extended by a few method steps.

25 In the second computer unit N, a new temporary identity variable TMUIN is formed for the first computer unit U and is subsequently allocated to the first computer unit U. This may be performed, for example, by 30 generating a random number or by means of tables in which potential identity variables are stored. The new temporary identity variable TMUIN for the first computer unit U is used to form a fifth encrypted term VT5 in the second computer unit N by encrypting the new 35 temporary identity variable TMUIN for the first computer unit U with the session key K using the encryption function Enc.

In this case, the second message M2 additionally contains at least the fifth encrypted term VT5. The fifth encrypted term VT5 is then decrypted in the first computer unit U. The new temporary identity variable
5 TMUIN for the first computer unit U is now available in the first computer unit U.

So that the second computer unit N is also assured of the fact that the first computer unit U has received
10 the new temporary identity variable TMUIN correctly, the third input variable for the first hash function h1 or for the second hash function h2 additionally contains at least the new temporary identity variable TMUIN for the first computer unit U.
15

A few alternatives to the exemplary embodiments described above are illustrated below:

The invention is not restricted to a mobile radio system, and hence it is also not restricted to a user of a mobile radio system and to the network, but rather may be used in all areas in which cryptographic key interchange between two communication parties is required. This may be the case, for example, in a communication link between two computers wishing to interchange data in encrypted form. Without any restriction to the general validity, a first communication party was called the first computer unit U and a second communication party was called the
25 second computer unit N above.
30

The following publications were cited as part of this document:

- [1] A. Aziz, W. Diffie, "Privacy and Authentication for Wireless Local Area Networks", IEEE Personal Communications, 1994, pp. 25 to 31
- [2] M. Beller, "Proposed Authentication and Key Agreement Protocol for PCS", Joint Experts Meeting on Privacy and Authentication for Personal Communications, P&A JEM 1993, 1993, pp. 1 to 11
- [3] C. Carroll, Y. Frankel, Y. Tsounis, "Efficient key distribution for slow computing devices", Conference Security&Privacy, Oakland, 1998
- [4] J. Zhou, K. Lam, "Undeniable billing in mobile communications", preprint 1998
- [5] US 5 214 700
- [6] DE brochures: Telesec. Telekom, Produktentwicklung Telesec beim Fernmeldeamt Siegen [Telesec. Telecom, product development Telesec at the Siegen exchange], pp. 12-13
- [7] US 5 222 140
- [8] US 5 153 919
- [9] M. Beller et al, Privacy and Authentication on a Portable Communication System, IEEE Journal on Selected Areas in Communications, Vol. 11, No. 6, pp. 821-829, 1993
- [10] DE 195 18 5465 C1

- [11] W. Diffie, P. C. van Oorschot, M. Wiener,
"Authentication and authenticated key exchanges",
Designs, Codes and Cryptography, Vol. 2, pp. 107-
125, 1992